JBoss Enterprise Application Platform 5
JBoss Cache Frequently Asked Questions

---

for Use with JBoss Enterprise Application Platform 5
Edición 5.1.0

Ben Wang
Scott Marlow
Bela Ban
Galder Zamarreño
Manik Surtani

JBoss Enterprise Application Platform 5 JBoss Cache Frequently Asked Questions

---

for Use with JBoss Enterprise Application Platform 5
Edición 5.1.0

Ben Wang

Bela Ban

Manik Surtani

Scott Marlow

Galder Zamarreño

Edited by

Laura Bailey

Legal Notice

Copyright © 2011 Red Hat, Inc.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This book is a compilation of frequently asked questions about JBoss Cache for use with JBoss Enterprise Application Platform 5 and its patch releases.

Table of Contents

Preface **3**
1. Convenciones del Documento 3
1.1. Convenciones tipográficas 3
1.2. Convenciones del documento 4
1.3. Notas y Advertencias 5
2. Cómo obtener ayuda y hacer sus comentarios 5
2.1. ¿Necesita ayuda? 5
2.2. ¡Necesitamos sus comentarios! 6

Capítulo 1. Información general **7**

Capítulo 2. JBoss Cache: Core **8**

Capítulo 3. Políticas de eliminación **15**

Capítulo 4. Cargadores de caché **16**

Capítulo 5. Resolución de problemas **18**

Historial de revisiones **19**

Preface

1. Convenciones del Documento

Este manual utiliza varias convenciones para resaltar algunas palabras y frases y llamar la atención sobre ciertas partes específicas de información.

En ediciones PDF y de papel, este manual utiliza tipos de letra procedentes de Liberation Fonts. Liberation Fonts también se utilizan en ediciones de HTML si están instalados en su sistema. Si no, se muestran tipografías alternativas pero equivalentes. Nota: Red Hat Enterprise Linux 5 y siguientes incluyen Liberation Fonts predeterminadas.

1.1. Convenciones tipográficas

Se utilizan cuatro convenciones tipográficas para llamar la atención sobre palabras o frases específicas. Dichas convenciones y las circunstancias en que se aplican son las siguientes:

`Negrita monoespaciado`

Utilizado para resaltar la entrada del sistema, incluyendo los comandos de shell, nombres de archivos y rutas. También sirve para resaltar teclas y combinaciones de teclas. Por ejemplo:

> Para ver el contenido del archivo **`my_next_bestselling_novel`** en su directorio actual de trabajo, escriba el comando **`cat my_next_bestselling_novel`** en el intérprete de comandos de shell y pulse **`Enter`** para ejecutar el comando.

El ejemplo anterior incluye un nombre de archivo, un comando de shell y una tecla . Todo se presenta en negrita-monoespaciado y distinguible gracias al contexto.

Las combinaciones de teclas se pueden distinguir de las individuales con el signo más que conecta cada partee de la combinación de tecla. Por ejemplo:

> Pulse **`Enter`** para ejecutar el comando.
>
> Pulse **`Ctrl`**+**`Alt`**+**`F2`** para pasar a una terminal virtual.

El primer ejemplo resalta una tecla particular a pulsar. El segundo ejemplo, resalta una combinación de teclas: un set de tres teclas pulsadas simultáneamente.

Si se discute el código fuente, los nombres de las clase, los métodos, las funciones, los nombres de variables y valores de retorno mencionados dentro de un párrafo serán presentados en **`Negrita-monoespaciado`**. Por ejemplo:

> Las clases de archivo relacionadas incluyen **`filename`** para sistema de archivos, **`file`** para archivos y **`dir`** para directorios. Cada clase tiene su propio conjunto asociado de permisos.

Negrita proporcional

Esta denota palabras o frases encontradas en un sistema, incluyendo nombres de aplicación, texto de cuadro de diálogo, botones etiquetados, etiquetas de cajilla de verificación y botón de radio; títulos de menú y títulos del submenú. Por ejemplo:

> Seleccione **Sistema** → **Preferencias** → **Ratón** desde la barra del menú principal para lanzar **Preferencias de ratón**. En la pestaña de **`Botones`**, seleccione la cajilla de **`ratón`**

de mano izquierda y luego haga clic en **Cerrar** para cambiar el botón principal del ratón de la izquierda a la derecha (adecuando el ratón para la mano izquierda).

Para insertar un carácter especial en un archivo **gedit**, seleccione **Aplicaciones** → **Accesorios** → **Mapa de caracteres** de la barra del menú. Luego, seleccione **Búsqueda** → **Buscar...** de la barra del menú de **Mapa de caracteres**, escriba el nombre del carácter en el campo de **Búsqueda** y haga clic en **Siguiente**. El carácter que buscó será resaltado en la **Tabla de caracteres**. Haga doble clic en ese carácter resaltado para colocarlo en el campo de **Texto a copiar** y luego haga clic en el botón **Copiar**. Ahora regrese al documento y elija **Modificar** → **Pegar** de la barra de menú de **gedit**.

El texto anterior incluye nombres de aplicación; nombres y elementos del menú de todo el sistema; nombres de menú de aplicaciones específicas y botones y texto hallados dentro de una interfaz gráfica de usuario, todos presentados en negrita proporcional y distinguibles por contexto.

Itálicas-negrita monoespaciado o ***Itálicas-negrita proporcional***

Ya sea negrita monoespaciado o negrita proporcional, la adición de itálicas indica texto reemplazable o variable. Las itálicas denotan texto que usted no escribe literalmente o texto mostrado que cambia dependiendo de la circunstancia. Por ejemplo:

Para conectar a una máquina remota utilizando ssh, teclee **ssh *nombre de usuario@dominio.nombre*** en un intérprete de comandos de shell. Si la máquina remota es **example.com** y su nombre de usuario en esa máquina es john, teclee **ssh john@example.com**.

El comando **mount -o remount *file-system*** remonta el sistema de archivo llamado. Por ejemplo, para volver a montar el sistema de archivo **/home**, el comando es **mount -o remount /home**.

Para ver la versión de un paquete actualmente instalado, utilice el comando **rpm -q *paquete***. Éste entregará el resultado siguiente: ***paquete-versión-lanzamiento***.

Observe que las palabras resaltadas en itálicas — nombre de usuario, dominio.nombre, sistema de archivo, paquete, versión y lanzamiento. Cada palabra es un marcador de posición, ya sea de texto a ingresar cuando se ejecuta un comando o para un texto ejecutado por el sistema.

Aparte del uso estándar para presentar el título de un trabajo, las itálicas denotan el primer uso de un término nuevo e importante. Por ejemplo:

Publican es un sistema de publicación de *DocBook*.

1.2. Convenciones del documento

Los mensajes de salida de la terminal o fragmentos de código fuente se distinguen visualmente del texto circundante.

Los mensajes de salida enviados a una terminal se muestran en **romano monoespaciado** y se presentan así:

```
books        Desktop   documentation  drafts  mss    photos   stuff  svn
books_tests  Desktop1  downloads      images  notes  scripts  svgs
```

Los listados de código fuente también se muestran en **romano monoespaciado**, pero se presentan y resaltan de la siguiente manera:

```
package org.jboss.book.jca.ex1;

import javax.naming.InitialContext;

public class ExClient
{
   public static void main(String args[])
       throws Exception
   {
      InitialContext iniCtx = new InitialContext();
      Object         ref    = iniCtx.lookup("EchoBean");
      EchoHome       home   = (EchoHome) ref;
      Echo           echo   = home.create();

      System.out.println("Created Echo");

      System.out.println("Echo.echo('Hello') = " + echo.echo("Hello"));
   }
}
```

1.3. Notas y Advertencias

Finalmente, utilizamos tres estilos visuales para llamar la atención sobre la información que de otro modo se podría pasar por alto.

Nota

Una nota es una sugerencia, atajo o enfoque alternativo para una tarea determinada. Ignorar una nota no debería tener consecuencias negativas, pero podría perderse de algunos trucos que pueden facilitarle las cosas.

Importante

Los cuadros con el título de importante dan detalles de cosas que se pueden pasar por alto fácilmente: cambios de configuración únicamente aplicables a la sesión actual, o servicios que necesitan reiniciarse antes de que se aplique una actualización. Ignorar estos cuadros no ocasionará pérdida de datos, pero puede causar enfado y frustración.

Aviso

Las advertencias no deben ignorarse. Ignorarlas muy probablemente ocasionará pérdida de datos.

2. Cómo obtener ayuda y hacer sus comentarios

2.1. ¿Necesita ayuda?

Si encuentra dificultades con alguno de los procedimientos descritos en este documento, visite el Portal del cliente de Red Hat en http://access.redhat.com. A través del portal del cliente, usted podrá:

- buscar o navegar a través de la base de artículos de soporte técnico sobre productos de Red Hat.
- enviar un caso de soporte a Servicios de Soporte Global de Red Hat (GSS)
- acceder a otra documentación del producto.

Red Hat alberga una lista grande de correos electrónicos para discutir sobre software de Red Hat y tecnología. Encontrará un listado de las listas de correo disponibles al público en https://www.redhat.com/mailman/listinfo. Haga clic en el nombre de la lista a la que quiera suscribirse o para acceder a los archivos de listados.

2.2. ¡Necesitamos sus comentarios!

Si encuentra algun error o si se le ocurre una manera de mejorar este manual, nos encantaría escuchar sus sugerencias. Complete un reporte en Bugzilla frente al producto **JBoss Enterprise Application Platform 5** y el componente **doc-Cache_FAQ**. El siguiente enlace le llevará a un reporte de error ya completado para este producto: http://bugzilla.redhat.com/.

Llene la siguiente plantilla en el campo de **Description** de Bugzilla. Sea tan especifico como le sea posible al describir el problema, esto ayudará a asegurarnos de que lo podemos solucionar rápidamente.

```
URL del documento:


Número de la sección y nombre:


Describa el problema:


Sugerencias para mejorar:


Información adicional:
```

Asegúrese de darnos su nombre para poder darle todo el crédito por reportar el problema.

Capítulo 1. Información general

P: **¿Qué es JBoss Cache?**

R: JBoss Cache es un caché transaccional y replicado. Es replicado ya que se pueden distribuir múltiples instancias de JBoss Cache (ya sea dentro de la misma MVJ o a través de varias MVJ ya sea que se encuentren en la misma máquina o en diferentes máquinas en una red) y los datos se replican a través de todo el grupo. Es transaccional ya que el usuario puede configurar un administrador de transacciones que cumpla con los requerimientos de JTA y hacer cualquier interacción del caché transaccional y los cachés participarían en transacciones JTA en curso. Observe que el caché también se puede ejecutar sin ninguna replicación, este es el modo local.

JBoss Cache comes in two flavours: Core and POJO versions. The core library (using the **`org.jboss.cache.Cache`** interface) is the underlying library that organises data in a tree-like structure and handles all locking, passivation, eviction and replication characteristics of data in the cache. The POJO library (using the **`org.jboss.cache.pojo.PojoCache`** interface) is built atop the core library and allows introspection of objects in the cache providing transparent coherence by using JBoss AOP. Note that the POJO edition of JBoss Cache (often referred to as POJO Cache) comes with a separate set of documentation (*POJO Cache User Guide*, FAQ, etc.) available on the JBoss Cache documentation website.

P: **¿Quiénes son los desarrolladores de JBoss Cache?**

R: JBoss Cache cuenta con una comunidad activa de desarrolladores y contribuidores. El proyecto lo fundó Bela Ban y actualmente lo lidera Manik Surtani. Jason Greene es el líder para el subsistema POJO Cache y otros de los contribuidores son Ben Wang, Harald Gliebe, Brian Stansberry, Vladimir Blagojevic, Mircea Markus, Jimmy Wilson, Galder Zamarreño y Elias Ross.

P: **¿Cómo saber que versión de JBoss Cache estoy utilizando?**

R: **`java -jar jbosscache-core.jar`** imprimirá los detalles de la versión.

P: **¿Cómo puedo migrar mi aplicación y configuración de JBoss Cache 1.x a 2.x?**

R: Revise esta página wiki para obtener ayuda.

P: **¿Y desde 2.x a 3.x?**

R: JBoss Cache 3.x tiene una API compatible con 2.x, aunque en tanto sea posible debe refactorizar su código para no utilizar métodos que ya no están en uso ya que es posible que estos desaparezcan en lanzamientos futuros de JBoss Cache.

JBoss Cache 3.x viene junto con un formato de configuración completamente nuevo. Los antiguos archivos de configuración 2.x todavía funcionarán aunque recibirá una advertencia en los registros con referencia a esto. De nuevo, en tanto sea posible, le recomendamos migrar su archivo de configuración al nuevo formato. Los scripts vienen junto con la distribución JBoss Cache 3.x para migrar los archivos de configuración (consulte **`config2to3.sh`** y **`config2to3.bat`**).

Observe que para tomar ventaja de algunas de las nuevas funcionalidades en JBoss Cache 3.x necesita utilizar el nuevo formato de configuración.

Capítulo 2. JBoss Cache: Core

P: **¿Puedo ejecutar múltiples instancias de JBoss Cache en la misma MV?**

R: Sí. Hay varios escenarios en donde podría ejecutar múltiples instacias de JBoss Cache. Por ejemplo, quiere ejecutar múltiples instancias caché locales en donde cada instancia tiene su propia configuración (por ejemplo, un política de caché diferente). En este caso, necesitará múltiples archivos de configuración xml.

P: **¿JBoss Cache puede ejecutar como un caché de segundo nivel dentro de Hibernate?**

R: Sí. Desde el lanzamiento de Hibernate 3.0 puede configurarlo para que utilice JBoss Cache como un caché de segundo nivel. Para obtener más detalles consulte la documentación de Hibernate y también refiérase a esta página wiki.

JBoss Cache 3.x junto con MVCC en particular funciona bastante bien como un caché de segundo nivel Hibernate.

P: **¿Qué tal utilizar POJO Cache como un caché Hibernate?**

R: No es necesario utilizar POJO Cache para un caché de segundo nivel dentro de Hibernate ya que Hibernate administra campos detallados en objetos Java. Así que el utilizar PojoCache no proporcionará ninguna ventaja y representará una desventaja innecesaria en el rendimiento.

P: **¿Cómo puedo configurar JBoss Cache?**

R: You can configure the JBoss Cache through a configuration XML file or programmatically using a **`org.jboss.cache.config.Configuration`** object, passed in to the **`org.jboss.cache.CacheFactory`** instance.

P: **¿Puedo utilizar un esquema o un DTD para validar mi archivo de configuración JBoss Cache?**

R: As of JBoss Cache 3.x, yes. An XSD schema is provided in your jbosscache-core.jar file, and is also available online, on http://www.jboss.org/jbosscache/jbosscache-config-3.0.xsd. You can configure your IDE, text editor or XML authoring tool to use this schema to validate your file.

P: **¿Cuál es la diferencia entre los diferentes modos caché?**

R: JBossCache has five different cache modes, i.e., **`LOCAL`**, **`REPL_SYNC`**, **`REPL_ASYNC`**, **`INVALIDATION_SYNC`** and **`INVALIDATION_ASYNC`**. If you want to run JBoss Cache as a single instance, then you should set the cache mode to **`LOCAL`** so that it won't attempt to replicate anything. If you want to have synchronous replication among different JBoss Cache instances, you set it to **`REPL_SYNC`**. For asynchronous replication, use **`AYSNC_REPL`**. If you do not wish to replicate cached data but simply inform other caches in a cluster that data under specific addresses are now stale and should be evicted from memory, use **`INVALIDATION_SYNC`** or **`INVALIDTAION_ASYNC`**. Synchronous and asynchronous behavior applies to invalidation as well as replication.

Observe que **`REPL_ASYNC`** y **`INVALIDATION_ASYNC`** no-bloquean. Esto puede ser útil cuando quiere tener otro JBoss Cache como espejo o copia de seguridad y no quiere esperar la confirmación de que este espejo ha recibido sus mensajes.

P: ¿Cómo funciona el mecanismo de replicación de JBoss Cache?

R: JBoss Cache influencia JGroups para las comunicaciones de red. Una sección de configuración JGroups está presente en su configuración JBoss Cache .

Un usuario puede configurar el clúster de instancias de JBoss Cache al compartir el mismo nombre de clúster (**`cluster name`**). También hay una opción de popular los datos en caché a iniciar una nueva instancia en el atributo **`ClusterConfig`**.

Note that once all instances join the same replication group, every replication change is propagated to all participating members. There is no mechanism for sub-partitioning where some replication can be done within only a subset of members, unless you use the Buddy Replication features. See the *JBoss Cache User Guide* for more details on this.

P: Tengo un clúster de 2 nodos. Si la red se muere ¿los cachés continuan ejecutando?

R: Sí, ambos continuarán ejecutando, pero dependiendo de su modo de replicación, puede que no todas las transacciones u operaciones se completen. Si se utiliza **`REPL_SYNC`** las operaciones fallarán mientras que si utiliza **`REPL_ASYNC`** las operaciones serán exitosas. Incluso si tienen exito, los cachés no estarán sincronizados.

P: ¿Puedo conectar la biblioteca X en vez de JGroups para manejar llamadas remotas y la comunicación de grupos?

R: En este punto la respuesta es no. Sí contamos con una capa de abstracción entre la suite de comunicación y JBoss Cache en las tuberías y puede que esta funcionalidad se agregue en algún momento en el futuro.

P: ¿El caché necesita replicarse en todas las instancias en el clúster? ¿Esto no sería muy lento si el clúster es grande?

R: No es necesario que la replicación tenga lugar en todos los nodos en el clúster. Esta funcionalidad - llamada replicación de compañeros - le permite a cada nodo seleccionar uno o más 'compañeros' en el clúster y sólo replica en su compañeros. Esto le permite al clúster escalar fácislmente sin un impacto extra en la memoria o en el tráfico de red con cada nodo añadido.

Consulte el manual del usuario para obtener mayor información sobre la replicación de compañeros y la manera que se puede utilizar para lograr una alta escalabilidad.

P: Estoy utilizando la replicación de compañeros ¿Necesito tener alguna forma de afinidad de sesión?

R: La afinidad de sesión se relaciona con el volver a la misma instancia caché para los mismos datos que se están utilizando. Aunque esto no es un requerimiento para la replicación de compañeros sí se recomienda bastante con el fin de minimizar el tener que mover el estado alrededor del clúster.

P: If I have the need for different configuration properties (e.g., `CacheMode` and `IsolationLevel`), do I simply need to create multiple `org.jboss.cache.Cache` instances with the appropriate configuration?

R: Sí. Todas las porpiedades mencionadas anteriormente son por cada instancia de caché. Por lo

tanto necesitará una instancia **`org.jboss.cache.Cache`** por separado.

P: ¿Esto no es costoso desde el punto de vista de la red, es decir el tener que crear sockets para cada instancia `org.jboss.cache.Cache`?

R: Sí puede llegar a ser costoso. Para dichos casos se recomienda que configure su caché usando el JGroups Multiplexer, el cual permite que varios cachés compartan un solo canal JGroups. Consulte el manual del usuario para obtener mayores detalles sobre cómo configurar el JGroups Multiplexer.

Un enfoque más rápido y eficiente es utilizar un transporte compartido en JGroups. Consulte la documentación de JGroups para obtener mayores detalles sobre cómo lograr esto.

P: El elemento de configuración `ClusterName` tiene alguna relación con el `PartitionName` del clúster de JBoss AS?

R: Sí. Ambos son nombres de grupo JGroups. Además de la noción de canal en JGroups, también puede realizar particiones en el canal en diferentes nombres de grupos.

P: When using multiple JGroups based components (`cluster-service.xml`, cache [multiple instances]), what is the correct/valid way to configure those components to make sure my multicast addresses don't conflict?

R: Hay dos parámetros que se deben tomar en consideración: dirección multicast (más el puerto) y el nombre del grupo. Como mínimo tendrá que ejecutar componentes utilizando un nombre de grupo diferente. El ejecutarlos en el mismo canal depende de si el rendimiento de la comunicación es crítico para usted o no. Si sí lo es entonces sería mejor ejecutarlos en canales diferentes.

P: ¿JBoss Cache soporta el almacenamiento de persistencia en caché?

R: Sí. JBoss Cache tiene una interfaz cargadora de caché que soporta la persistencia del caché. A continuación encontrará más preguntas frecuentes sobre los cargadores de caché.

P: ¿JBoss Cache soporta el caché pasivo/en exceso en un almacenamiento de datos?

R: Sí. JBoss Cache utiliza el cargador de caché para soportar el caché pasivo/en exceso. Consulte la documentación para aprender cómo configurar y utilizar esta funcionalidad.

P: ¿JBoss Cache es seguro con hilos?

R: Sí, es seguro con hilos.

P: ¿Ahora JBoss Cache soporta transacciones XA (2PC)?

R: No, aunque también se encuentra en nuestra lista de cosas por hacer. Nuestra implementación interna utiliza un procedimiento 2PC similar para coordinar una transacción entre diferentes instancias, pero JBoss Cache no es un recurso XA.

P: ¿Cuáles administradores de transacciones son soportados por JBoss Cache?

R: JBoss Cache soporta cualquier TransactionManager que cumpla con los requerimientos de JTA tal como JBoss Transactions.

Aunque JBoss Cache se envía junto con un administrador de transacciones de mentiras (**`org.jboss.cache.transaction.DummyTransactionManager`**) *no* recomendamos el utilizarlo para producción. No es seguro para hilos y es destinado sólamente para pruebas internas.

P: ¿Cómo configuro el caché para que sea transaccional?

R: Puede utilizar el administrador de transacciones predeterminado que se envía junto con JBoss AS o tiene que implementar la interfaz **`org.jboss.cache.transaction.TransactionManagerLookup`** y retornar una instancia de su implementación **`javax.transaction.TransactionManager`**. La propiedad de configuración **`TransactionManagerLookupClass`** define la clase que el caché debe utilizar para buscar una referencia en un administrador de transacciones. Es trivial el implementar esta clase para soportar otros administradores de transacciones. Una vez este atributo se especifique entonces el caché buscará el contexto de transacción desde este administrador de transacciones.

La clase **`org.jboss.cache.transaction.GenericTransactionManagerLookup`** que se envía junto con JBoss Cache puede detectar y enlazar los administradores de transacciones más populares. Consulte los Javadocs **`GenericTransactionManagerLookup`** para obtener mayor información.

P: How do I control the Cache locking level?

R: JBoss Cache lets you control the cache locking level through the transaction isolation level. This is configured through the attribute **`IsolationLevel`**. The transaction isolation levels correspond to database isolation levels, namely **`NONE`**, **`READ_UNCOMMITTED`**, **`READ_COMMITTED`**, **`REPEATABLE_READ`**, and **`SERIALIZABLE`**. Note that these isolation levels are ignored if optimistic locking is used. For details, please refer to the *JBoss Cache User Guide*.

Desde JBoss Cache 3.x, al utilizar el esquema de bloqueo MVCC sólo se soportan **`READ_COMMITTED`** y **`REPEATABLE_READ`**. Cualquier nivel de aislamiento se ajustará así mismo.

P: ¿Cómo hace JBoss Cache para bloquear datos para acceso simultáneo?

R: Por defecto en JBoss Cache 2.x se utiliza el bloqueo pesimista para bloquear nodos de datos con base en el nivel de aislamiento configurado. También ofrecemos el bloqueo optimista para permitir una mayor simultaneidad con un pequeño sobrecosto en el procesamiento y en el rendimeinto. Consulte la documentación y allí encontrará una discusión más detallada sobre concurrencia y bloqueo en JBoss Cache.

En JBoss Cache 3.x, el bloque optimista y pesimista ya no se utilizan a favor de MVCC (del inglés multi-version concurrency control -control de concurrencia multi-versión), el cual es mucho más eficiente que el bloqueo optimista o pesimista. Para ver una discusión más detallada sobre nuestra implementación MVCC consulte este blog y esta página wiki.

P: ¿Cómo habilito el bloqueo optimista o MVCC en JBoss Cache?

R: Please see the configuration section of the *JBoss Cache User Guide* for details.

P: ¿Puedo utilizar el nivel de bloqueo de caché incluso sin un contexto transaccional?

R: Sí. JBossCache controla el comportamiento de bloqueo individual de nodos por medio de la semántica a nivel de aislamiento. Esto significa que incluso si no utiliza una transacción puede especificar el nivel de bloqueo por medio del nivel de aislamiento. Puede considerar el comportamiento de bloqueo de nodos por fuera de una transacción como si se encontrara bajo una transacción con el **`auto_commit`** funcionando.

P: ¿JBoss Cache soporta la semántica `SELECT FOR UPDATE`?

R: Sí, pero solo es posible si está ejecutando dentro de una transacción JTA *y* si está utilizando **`MVCC`** o **`PESSIMISTIC`**como esquema de bloqueo de nodos.

Para lograr la semántica **`SELECT FOR UPDATE`** simplemente haga:

```
// start transaction ...

cache.getInvocationContext().getOptionOverrides().setForceWriteLock(true);
Node n = cache.get("/a/b/c"); // this acquires a WRITE LOCK on this node
    ...
    ...

// end transaction
```

P: With replication (`REPL_SYNC`/`REPL_ASYNC`) or invalidation (`INVALIDATION_SYNC`/`INVALIDATION_ASYNC`), how often does the cache broadcast messages over the network?

R: Si las actualizaciones se encuentran bajo una transacción entonces las emisiones sólo tienen lugar cuando la transacción está a punto de guardar los cambios (de hecho durante la fase de preparación internamente). Es decir, será una actualización en grupo. Sin embargo, si las operaciones no se encuentran bajo el contexto de transacción entonces cada actualización disparará la replicación. Observe que esto tiene implicaciones en el rendimiento si la latencia de red representa un problema.

P: ¿Cómo puedo realizar una eliminación en masa?

R: Si realiza **`cache.removeNode("/myroot")`**, esto eliminará recursivamente todas las entradas bajo "/myroot".

P: ¿Puedo monitorear y administrar el JBoss Cache?

R: Yes, using a JMX console such as the one shipped with JBoss AS or JDK 5's **`jconsole`** utility. See the chapter titled **Management Information** in the *JBoss Cache User Guide* for more details.

P: JBoss Cache usa un caracter ":" en su nombre de objeto. Esto causa problemas con mi servidor MBean. ¿Qué puedo hacer?

R: This is something we have seen with some MBean servers. By default, JBoss Cache uses **`jboss.cache:service=JBossCache`** as a prefix to all objects it binds in JMX. To work around this, use the **`-Djbosscache.jmx.prefix`** JVM parameter to pass in an alternate prefix.

P: ¿Puedo deshabilitar los atributos de administración de JBoss Cache?

R: Yes, you can. See the section on configuration in the *JBoss Cache User Guide*.

P: ¿Qué pasó con jboss-serialization.jar?

R: Desde JBoss Cache 2.0.0, la dependencia de JBoss Serialization ya no se utiliza ya que la mayoría de los beneficios de JBoss Serialization se encuentran disponibles en las máquinas virtuales actualizadas Java 5. Ya que JBoss Cache 2.0.0 tiene los valores iniciales con base en Java 5 no había necesidad de proporcionar estos beneficios separadamente.

P: ¿JBoss Cache soporta particionamientos?

R: Por ahora no. JBoss Cache no soporta particionamientos que un usuario puede configurar para tener diferentes grupos de datos residiendo en diferentes instancias caché y al mismo tiempo participando como un grupo de replicación.

P: ¿JBoss Cache maneja el concepto de la aplicación de carga de clase dentro de digamos, un contendor Java EE?

R: La carga de clase especifica a la aplicación se utiliza bastante dentro de un contenedor Java EE. Por ejemplo, una aplicación web puede requerir un nuevo cargador de clase para poner en ámbito una versión especifica de la biblioteca del usuario. Sin embargo, por defecto JBoss Cache es agnóstico para el cargador de clase. En general, esto conlleva dos tipos de problemas:

- La instancia del objeto se almacena en el cache1 y la réplica en cache2. Como consecuencia, el cargador de clase del sistema crea la instancia en cache2. Puede que la replicación falle si el cargador de clase del sistema en cache2 no tiene acceso a la clase requerida. Inclusive si la replicación no falla, puede que un hilo de usuario en cache2 no pueda acceder al objeto si el hilo del usuario está esperando un tipo definido por el cargador de clase.
- La instancia del objeto es creada por el hilo 1 y será accedida por el hilo 2 (con dos cargadores de clase diferentes). JBoss Cache no tiene noción de los diferentes cargadores de clase involucrados. Como consecuencia, se presentará una **`ClassCastException`**. Este es un problema estándar al pasar un objeto del espacio de una aplicación a otro; JBoss Cache simplemente añade un nivel de indirección al pasar el objeto.

To solve the first kind of issue JBoss Cache uses a **`CacheMarshaller`**. Basically, this allows application code to register a classloader with a portion of the cache tree for use in handling objects replicated to that portion. See the **`CacheMarshaller`** section of the *JBoss Cache User Guide* for more details.

Para resolver la segunda clase de problema puede utilizar la opción de configuración **`UseLazyDeserialization`** en JBoss Cache, la cual envuelve sus objetos en un wrapper **`Marshalledvalue`**. El **`MarshalledValue`** serializa y des-serializa su objeto cuando lo necesite, asegurándose de que se utilice cada vez el cargador de clase del contexto local del hilo apropiado.

P: ¿Actualmente JBoss Cache soporta la notificación de pre-eventos y post-eventos?

R: Sí. Se pasa un valor boolean a cada callback de notificación identificando si el callback se presentó antes o despues del evento. Consulte la anotación **`org.jboss.cache.notifications.annotations.CacheListener`** para obtener mayores

detalles.

P: ¿Cómo puedo implementar un escucha personalizado para que escuche los eventos caché?

R: See the *JBoss Cache User Guide* on this subject.

P: Can I use the `UseRegionBasedMarshalling` attribute in JBoss Cache in order to get around ClassCastExceptions happening when accessing data in the cache that has just been redeployed?

R: Sí, si puede. Originalmente, cache Marshalling fue diseñado como una solución opcional para aquellos cachés replicados que en estado de transferencia no tenían acceso a los cargadores de clase que definen los objetos en el caché.

On each deployment, JBoss creates a new classloader per the top level deployment artifact, for example an EAR. You also have to bear in mind that a class in an application server is defined not only by the class name but also its classloader. So, assuming that the cache is not deployed as part of your deployment, you could deploy an application and put instances of classes belonging to this deployment inside the cache. If you did a redeployment and try to do a get operation of the data previously put, this would result in a ClassCastException. This is because even though the class names are the same, the class definitions are not. The current classloader is different to the one when the classes were originally put.

Al habilitar el marshalling, puede controlar el ciclo de vida de los datos en el caché o si va a borrar la implementación puede inactivar la región y des-registrar el cargador de clase que habría registrado en el despliegue y sacaría los datos en el caché localmente. Eso significa que en el próximo despliegue, los datos no estarán en el caché y por lo tanto se evitará el problema. Obviamente, el utilizar marshalling para evitar este problema sólo se recomienda cuando tenga alguna clase de persistencia respaldando en donde los datos sobreviven, por ejemplo, utilizando CacheLoaders o cuando JBoss Cache se utiliza como caché de segundo nivel en un marco de trabajo de persistencia.

Para implementar esta funcionalidad siga las instrucciones indicadas en el ejemplo que se encuentra en la sección CacheMarshaller del manual del usuario. Vale la pena anotar que en vez de un **`ServletContextListener`**, podría añadir este códido en un **`MBean`** que contenía métodos de ciclos de vida tal como **`start()`** y **`stop()`**. La clave sería que este MBean dependiera del caché destino de manera que pueda operar en tanto el caché se encuentre funcionando.

Capítulo 3. Políticas de eliminación

P: ¿JBoss Cache soporta políticas de eliminación?

R: Yes. JBoss Cache currently supports multiple eviction policies such as LRU, MRU, and FIFO. Users can also plug in their own eviction policy algorithms. See the *JBoss Cache User Guide* for details.

P: Does JBoss Cache's eviction policy operate in replication mode?

R: Sí y no.

La política de eliminación sólo opera en modo local. Es decir que los nodos sólo se eliminan localmente. Esto puede hacer que el contenido del caché no esté sincronizado temporalmente. Pero cuando un usuario trata de obtener el contenido en caché de un nodo eliminado y se encuentre con que es nulo (por ejemplo, **`get`** retorna nulo), debe obtenerlo de la otra fuente de datos y debe repopular los datos en el caché. En este momento, el contenido del nodo será propagado y el contenido del caché estará sincronizado.

Sin embargo, todavía puede ejecutar políticas de eliminacióncon el modo caché configurado como **`REPL_SYNC`** o **`REPL_ASYNC`** . Dependiendo de su caso, puede configurar múltiples instancias caché para que tengan su propia política de eliminación (las cuales se aplican localmente) o simplemente tenga instancias seleccionadas con políticas de eliminación activadas.

También observe que con la opción de cargador caché, un nodo eliminado localmente también puede ser persistido en el almacenamiento backend y un usuario puede recuperarlo del almacenamiento después.

P: ¿JBoss Cache soporta `Region` ?

R: Sí. JBoss Cache cuenta con la noción de región en donde un usuario puede configurar los parámetros de la política de eliminación (por ejemplo, **`maxNodes`** o **`timeToIdleSeconds`**)

A region in JBoss Cache denotes a portion of tree hierarchy, e.g., a fully qualified name (**`org.jboss.cache.Fqn`**). For example, a user can define **`/org/jboss`** and **`/org/foocom`** as two separate regions. But note that you can configure the region programmatically now, i.e., everything has to be configured through the XML file.

P: I have turned on the eviction policy. Why do I still get an Out of Memory (OOM) exception?

R: OOM puede tener lugar cuando la velocidad del acceso al caché excede la velocidad del temporizador de manejo de la política de eliminación. El manejador de la política de eliminación se despertará cada **`wakeUpInterval`** milisegundos (o **`wakeUpIntervalSeconds`** segundos antes de 3.x) para procesar la cola del evento de eliminación. Así que cuando el tamaño de la cola esté lleno, creará un backlog y presentará excepciones fuera de memoria a menos de que el temporizador de eliminación lo alcance. Para solucionar este problema, además de incrementar el tamaño heap de la MV también puede reducir el **`wakeUpInterval`** de manera que el hilo del temporizador procese la cola con más frecunecia.

Capítulo 4. Cargadores de caché

P: ¿Qué es un cargador de caché?

R: Un cargador de caché es la conección de JBoss Cache a un almacenamiento de datos (persistente). JBoss Cache llama al cargador de caché para buscar datos desde un almacenamiento cuando esos datos no se encuentran en el caché y cuando las modificaciones se realizan en los datos en el caché se llama al cargador de caché para almacenar esas modificaciones en el almacenamiento.

Junto con las políticas de eliminación, JBoss Cache con un cargador de caché le permite a un usuario el mantener un caché vinculado para un gran almacenamiento de datos backend. Los datos usados con frecuencia se buscan desde el almacenamiento de datos en el caché y los datos menos utilizados se eliminan con el fin de proporcionar un rápido acceso a los datos accedidos con bastante frecuencia. Todo esto se configura por medio de XML y el programador no tiene que encargarse de cargar y eliminar.

JBoss Cache actualmente se envía junto con varias implementaciones del cargador de caché, incluyendo:

- **`org.jboss.cache.loader.FileCacheLoader`**: this implementation uses the file system to store and retrieve data. JBoss Cache nodes are mapped to directories, subnodes to subdirectories, etc. Attributes of a node are mapped to a data file inside the directory.
- **`org.jboss.cache.loader.jdbm.JdbmCacheLoader`**: this implementation is based on JDBM, an open source file-based transactional persistence engine.
- **`org.jboss.cache.loader.bdbje.BdbjeCacheLoader`**: this implementation is based on Oracle's Berkeley DB Java Edition database, a fast and efficient transactional database. It uses a single file for the entire store. Note that if you use the Berkeley DB cache loader with JBoss Cache and wish to ship your product, you will have to acquire a commercial license from Oracle.
- **`org.jboss.cache.loader.JDBCCacheLoader`**: this implementation uses the relational database as the persistent storage.
- And more. See the chapter on cache loaders in the *JBoss Cache User Guide* for more details.

P: ¿Se recomienda el FileCacheLoader para uso en producción?

R: No, no se recomienda. El FileCacheLoader tiene grandes limitaciones, las cuales restringen su uso en un entorno de producción o si se utiliza en dicho entorno entonces se debe utilizar con el debido cuidado y teniendo en cuenta dichas limitaciones.

- Debido a la manera en que el FileCacheLoader representa una estructura de árbol en el disco (directorios y archivos) el atraversar es poco eficiente para árboles grandes.
- Se debe evitar el uso en sistemas de archivos compartidos tal como NFS, Windows shares, etc ya que estos no implementan el bloqueo de archivos apropiado y pueden dañar datos.
- El uso con un nivel de aislamiento de NONE puede generar escrituras erróneas ya que múltiples hilos intentan escribir en el mismo archivo.
- Los sistemas de archivos son intrínsecamente no transaccionales así que al tratar de utilizar su caché en un contexto transaccional, los fallos al escribir en el archivo (lo cual tiene lugar durante la fase de guardado de los cambios) no se pueden recuperar.

Siempre se recomienda que el FileCacheLoader no se utilice en un entorno altamente transaccional, simultáneo o estresante y su uso se restringe para las pruebas.

P: ¿La escritura en cargadores de caché puede ser asincrónica?

R: Yes. Set the **async** attribute to true. See the JBoss Cache Users' Guide for a more detailed discussion. By default though, all cache loader writes are synchronous and will block.

P: ¿Puedo escribir mi propio cargador de caché?

R: Yes. A cache loader is a class implementing **`org.jboss.cache.loader.CacheLoader`** or extending **`org.jboss.cache.loader.AbstractCacheLoader`**. It is configured via the XML file (see *JBoss Cache User Guide*).

P: ¿Un cargador de caché tiene que utilizar un almacenamiento persistente?

R: No. A cache loader could, for example, fetch (and possibly store) its data from a webdav-capable webserver. Another example is a caching proxy server, which fetches contents from the web. Note that an implementation of CacheLoader may not implement the 'store' functionality in this case, but just the 'load' functionality.

P: ¿Puedo utilizar más de un cargador de caché?

R: Yes. Within the CacheLoaderConfiguration XML element (see the *JBoss Cache User Guide* chapter on cache loaders) you can describe several cache loaders. The impact is that the cache will look at all of the cache loaders in the order they've been configured, until it finds a valid, non-null element of data. When performing writes, all cache loaders are written to (except if the ignoreModifications element has been set to true for a specific cache loader).

P: ¿Puerdo migrar un almacenamiento de caché basado en JDBCacheLoader o FileCacheLoader que contenga datos formateado con JBoss Cache 1.x.x al formato de JBoss Cache 2.0?

R: Yes. See the "Transforming Cache Loaders" section within the "Cache Loaders" section located in the *JBoss Cache User Guide*.

P: ¿ El TCPDelegatingCacheLoader es resistente a re-inicios del TCPCacheServer?

R: As of JBoss Cache 2.1.0, the answer is yes. See the *JBoss Cache User Guide* for details on how to configure and tune your retries and wait period for reestablishing the TCP connection.

Antes de eso, el reiniciar el TCPCacheServer también significaría el reiniciar su aplicación que utiliza el caché.

Capítulo 5. Resolución de problemas

P: Tengo problemas en hacer funcionar JBoss Cache, ¿dónde puedo obtener información sobre resolución de problemas?

R: La sección sobre resolución de problemas se puede encontrar en el siguiente enlace wiki.

Historial de revisiones

Revisión 5.1.0-2.402	**Fri Oct 25 2013**	**Rüdiger Landmann**
Rebuild with Publican 4.0.0		
Revisión 5.1.0-2.1	**2013-06-11**	**Misty Stanley-Jones**
Rebuild for updated legal template		
Revisión 5.1.0-2	**2012-07-18**	**Anthony Towns**
Rebuild for Publican 3.0		
Revisión 5-1	**Wed Sep 15 2010**	**Laura Bailey**
Changed version number in line with new versioning requirements.		
Revised for JBoss Enterprise Application Platform 5.1.0.GA.		